PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor:	James E. King, et al.	Atty.Dkt.No.:	5681-85400
Serial Number:	10/774,755	Examiner:	Hussain, Tauqir
Filing Date:	February 9, 2004	Group/Art Unit:	2452
Title:	SWITCH	Conf. No.:	6135

******CERTIFICATE OF E-FILING TRANSMISSION******

I hereby certify that this correspondence is being transmitted via electronic filing to the United States Patent and Trademark Office on the date shown below:

On: April 5, 2010 /Dean M. Munyon/
Date Dean M. Munyon

APPEAL BRIEF

Further to the Final Office Action mailed November 4, 2009, Appellants present this Appeal Brief. **This Appeal Brief is timely filed; accordingly, no extension of time fee should be due.** Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

The real party in interest is Oracle America, Inc., formerly known as Sun Microsystems, Inc.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-68, 74, 75, 83-85, 91-92, 98-99, are canceled. Claims 69-73, 76-82, 86-90, 93-97 and 100-104 are pending in the case. Claims 69-73, 76-82, 86-90, 93-97, and 100-104 are finally rejected, and are the subject of this appeal. (A copy of these claims is found in the Claims Appendix.)

IV. STATUS OF AMENDMENTS

No amendments to the claims have been filed subsequent to the rejection in the Final Office Action of November 4, 2009.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The present application relates to the field of modular computer systems and, in some embodiments, to processing fault messages using a fault management unit included within a combined switch and service processor module. Independent claim 69 is directed to a combined switch and service processor module. Independent claim 82 is directed to a computer system. Independent claim 85 is a method of operating a combined switch and service processor module.

Independent claim 69 is directed to a combined switch and service processor module **(e.g., page 2, lines 3-5; CSSP 71 Figs. 7 and 9)** for a modular computer system **(e.g., page 2, lines 3-5)**. The combined switch and service processor module includes a switch portion **(e.g., switch 730 Figs. 7 and 9)**, a service processor portion **(e.g., service processor 740 Figs. 7 and 9)**, and a data interface configured to communicate with an external management entity via a network **(e.g., page 29, lines 9-14; UIS 701 Figs. 7 and 9)**. The combined switch and service processor portion also includes a fault management unit configured to receive fault messages generated by the switch portion and by the service processor portion **(e.g., page 30, lines 22-26)**, including respective fault messages generated by the switch portion and the service processor portion that relate to a common fault **(e.g., page 30, lines 18-22 and 31-33; page 31, line 1)**. The fault management unit is further configured to perform processing on a received fault message to determine whether to forward the received message to the external management entity via the data interface **(e.g., page 30, lines 24-26 and 28-30; page 31, lines 3-7)**. The fault management unit is configured to not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message to the external management entity **(e.g., page 30, lines 24-26 and 28-30; page 31, lines 3-7)**.

Independent claim 82 is directed to a computer system **(e.g., page 2, lines 3-5).** The computer system includes a combined switch and service processor module **(e.g., page 2, lines 3-5; CSSP 71 Figs. 7 and 9)** that includes a switch portion **(e.g., switch 730 Figs. 7 and 9)** and a service processor portion **(e.g., service processor 740 Figs. 7 and 9)**. The combined switch and service processor module includes a data interface

configured to communicate with an external management entity via a network **(e.g., page 29, lines 9-14; UIS 701 Figs. 7 and 9)**. The combined switch and service processor also includes a fault management unit configured to receive fault messages generated by the switch portion and by the service processor portion **(e.g., page 30, lines 22-26)**, including respective fault messages generated by the switch portion and the service processor portion that relate to a common fault **(e.g., page 30, lines 18-22 and 31-33; page 31, line 1)**. The fault management unit is further configured to perform processing on a received fault message to determine whether to forward the received message to the external management entity via the data interface **(e.g., page 30, lines 24-26 and 28-30; page 31, lines 3-7)**. The fault management unit is configured to not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message to the external management entity **(e.g., page 30, lines 24-26 and 28-30; page 31, lines 3-7)**.

Independent claim 86 is directed to a method of operating a combined switch and service processor module **(e.g., page 2, lines 3-5; CSSP 71 Figs. 7 and 9)** for a modular computer system **(e.g., page 2, lines 3-5)**. The combined switch and service processor module includes a switch portion **(e.g., switch 730 Figs. 7 and 9)**, a service processor portion **(e.g., service processor 740 Figs. 7 and 9)**, a data interface configured to communicate with an external management entity via a network **(e.g., page 29, lines 9-14; UIS 701 Figs. 7 and 9)**, and a fault management unit **(e.g., page 30, lines 22-26)**. The method includes the fault management unit receiving fault messages generated by the switch portion and by the service processor portion **(e.g., page 30, lines 22-26)** including a first fault message generated by the switch portion and a second fault message generated by the service processor portion, where the first and second fault messages relate to a common fault **(e.g., page 30, lines 18-22 and 31-33; page 31, line 1)**. The method further includes the fault management unit performing processing on a received fault message to determine whether to forward the received message to the external management entity via the data interface **(e.g., page 30, lines 24-26 and 28-30; page 31, lines 3-7)**. The method further includes the fault management unit not forwarding fault messages that relate to a fault for which the fault management unit has

already forwarded a fault message to the external management entity **(e.g., page 30, lines 24-26 and 28-30; page 31, lines 3-7)**.

Appellant notes that the above cites are provided as examples and are therefore not to be used to limit any scope to which Appellant is entitled.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1) Claims 69-73, 76-80, 82, 86-90, 93-97, 100, and 104 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwartz et al (U.S. Pub. No. 2005/0071625) in view of Garg et al (U.S. Pub. No. 2005/0193229).

2) Claim 81 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwartz in view of Garg and Ohkubo et al (U.S. Patent No. 5,276,683).

3) Claims 101, 102, and 103 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schwartz and Garg in view of Tzeng et al (U.S. Pub. No. 2005/0122825).

VII. ARGUMENT

1) Rejection Under 35 U.S.C. § 103(a) over Schwartz et al (U.S. Pub. No. 2005/0071625) in view of Garg et al (U.S. Pub. No. 2005/0193229)

Claims 69-73, 76-80, 82, 86-90, 93-97, 100, and 104 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwartz et al (U.S. Pub. No. 2005/0071625) in view of Garg et al (U.S. Pub. No. 2005/0193229). Final Office Action filed November 4, 2009 ("Final Office Action") at 2. Appellant traverses the rejections of these claims for at least the reasons set forth below.

The present disclosure is concerned, for example, with "a fault management unit" that is included within a "combined switch and service processor module for a modular computer system" (recited claim 69). The fault management unit is configured to receive "respective fault messages generated by [a] switch portion and [a] service processor portion that relate to a common fault" and to "not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message to [an] external management entity" (recited in claim 69). As will be evident from the discussion below, none of the cited references teach or suggest the "fault management unit" recited in the claims.

Overview of Schwartz

Schwartz is cited for teaching several of the features recited in the claims but is not alleged to teach or suggest the recited "fault management unit." *See* Final Office Action at 3 and 4.

Schwartz is directed to the "booting up and configuring multimode computer systems using a scalability management module." Schwartz 1:9-11. As Schwartz explains, "[t]he scalability management module sets and maintains configuration parameters for the multi-node computer." *Id.* at Abstract. "If one of the nodes is removed from the multi-node computer, a hot-spare node can be dynamically configured to replace

the removed node without having to reconfigur[e] or physically reconnect the remaining nodes." *Id.* To this end, Schwartz discloses that the "scalability management module (SMM) 212" includes a "master scalability chipset 210" and "each node 204" includes a "slave scalability chipset 208" storing "controller information." *Id.* at 2:50-58. When each computer node 204 boots up, "a single master scalability chipset may configure all slave scalability chipsets 208." *See id.* at 2:50-58 and 4:4:21.

Overview of Garg

Garg is cited for teaching a "fault management unit" and the various features associated with that unit. *See* Final Office Action at 3 and 4.

Garg is directed to "a software architecture that distributes processing load of an application among multiple processors" and that has "the ability to recover from multiple software and hardware failures in distributed systems." Garg ¶¶ [0001] and [0006]. As Garg explains, "[i]n a distributed processing system, a single software component or application executes in parallel on more than one processor," where "[e]ach copy of the executing application takes on some portion of the processing load." *Id.* ¶ [0124]. "Such applications may, for example, carry out complex scientific calculations or control essential services, such as city water purification or power grid control for a large population region." *Id.* at ¶[0004].

Garg refers to a "basic unit of load distribution" (i.e., "the portion of the processing load" sent to each processor) as being a "resource set," which is "a group of resources (such as messages, data, or network elements) that are used by the application to service external events." *Id.* ¶¶ [0036] and [0124]. Garg states that a problem with distributed processing systems is that "resource set[s]" are subject to "failure" and that "[t]he state of a resource set is specific to the processor on which it resides." *See id.* ¶¶ [0161] and [0162]. Garg discloses, however, that "[r]ecovery from the failure of an active resource set is possible if a corresponding standby resource set exists on some other processor." *Id.* ¶ [0162]. Stated another way, if an active resource of an application

fails, it is possible to recover from that failure by activating a corresponding resource that is on standby on another processor. *See id.*

Garg's software architecture includes a "System Controller" that "allow[s] the application to recover from multiple failures and redistribute incoming traffic on failures" by "activating the standby of all the failed resource sets," where the "new active copy takes over all processing from the failed active resource sets." *See id* at ¶¶ [0046] and [0392]. Garg describes this process of activating standby resource sets as a "forced switchover" or as the "API function" "scForcedSwitchover." *See id.* at ¶¶ [0046], [0156], and [0392].

When the system controller executes the scForcedSwitchover command, Garg discloses that "[i]f any of the [] steps of the scForcedSwitchover() command fail to complete successfully, the System Controller generates an alarm indicating the failure." *Id.* at ¶ [0405]. A "Fault Manager module uses this alarm to identify the location and cause of the failure." *Id.* at ¶ [0405]. "The Fault Manager isolates the new failure and typically issues a new scForcedSwitchover() command to the System Controller to recover from the new failure." *Id.* "This cycle continues until all failures have been recovered." *Id.* "On completion of the scForcedSwitchover() operation, the standby copy of the resource set becomes active." *Id.* at ¶¶ [0237] and [0406].

Claims 69-73, 76-80, 82, 86-90, 93-97, 100 and 104

Appellant traverses the rejection of claim 69 for at least the reasons set forth below. Claims 70-73, 76-80, 82, 86-90, 93-97, 100, and 104 stand or fall with representative claim 69, which recites as follows:

> 69. A combined switch and service processor module for a modular computer system, comprising:
>
> a switch portion;
>
> a service processor portion;
>
> a data interface configured to communicate with an external management entity via a network; and
>
> a fault management unit configured to receive fault messages generated by the switch portion and by the service processor portion, including respective fault messages generated by the switch portion and the service processor portion that relate to a common fault, and wherein the fault management unit is further configured to perform processing on a received fault message to determine whether to forward the received

message to the external management entity via the data interface, wherein the fault management unit is configured to not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message to the external management entity.

In response to the Office Action filed March 31, 2009, Appellant argued that Schwartz does not teach or suggest a "fault management unit" as recited in claim 69. *See* Office Action Response filed June 30, 2009 at 11. The Examiner agreed with Appellant's argument, noting that Schwartz "is silent on disclosing explicitly, a fault management unit." Final Office Action at 3. The Examiner instead cited Garg as teaching this feature. *Id.* Since Schwartz admittedly does not teach or suggest a "fault management unit," Garg must teach each and every feature of the "fault management unit" recited in claim 69 in order for the proposed combination of Schwartz and Garg to establish a *prima facie* case of obviousness with respect to that claim. *See* MPEP 2143.03. As set forth below, Garg does not teach or suggest several featured of the recited "fault management unit."

Garg Does Not Teach or Suggest "Receiv[ing] Fault Messages" that "Relate to a Common Fault" and "Not Forward[ing] [Related] Fault Messages"

Claim 69 recites "a fault management unit configured to receive fault messages," "including respective fault messages … that relate to a **common** fault" (emphasis added). Claim 69 further recites that the "fault management unit is further configured to perform processing on a received fault message to determine whether to forward the received message" and to "not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message." The Examiner cites "Garg, Fig. 17, element-Fault Manager, [0046]," and "[0405]" as teaching these features of claim 69. Final Office Action at 3 and 4. As described above, paragraph [0046] describes the "System Controller['s]" ability to "allow the application to recover from multiple failures" by "activating the standby of all the failed resource sets"—i.e., performing a "forced switchover." *See* Garg ¶ [0046] and [0156]. Paragraph [0405] describes how "the System Controller generates an alarm indicating [a] failure" to a "Fault Manager" "[i]f any of the [] steps of the scForcedSwitchover( ) command fail."

Appellant disagrees that these portions (or any other portions) of Garg teach or suggest the above-noted features of claim 69.

Appellant submits that Garg's ability to "recover from multiple failures" does not teach or suggest "respective fault messages ... that relate to a **common** fault," as recited in claim 69. The Examiner appears to assume that the "multiple failures" referred to in Garg necessarily relate to a "common fault"; however, the Examiner provides no specific support in Garg for this proposition. In fact, the Examiner has not provided evidence that Garg's system is even capable of generating "respective fault messages ... that relate to a common fault." Indeed, Garg's disclosure that its "Fault Manager" can "issue[] a new scForcedSwitchover( ) command to the System Controller to recover from the new failure" and that "[t]his cycle continues until all failures have been recovered," Garg at ¶ [0405], suggests that the "multiple failures" are distinct from one another, and thus *not* relate[d] to a "common fault" as in claim 69.

Even assuming *arguendo* that Garg teaches or suggests "respective fault messages ... that relate to a common fault," Appellant submits that Garg does not teach or suggest a "fault management unit [that] is further configured to perform processing on a received fault message to determine whether to forward the received message" and to "not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message," as recited in claim 69. As noted above, Garg is silent as to whether two or more failures can be generated "that relate to a common fault." Furthermore, Garg simply states that the "cycle" of issuing "scForcedSwitchover( )" commands to the System Controller "continues until all failures have been recovered," Garg at ¶ [0405]; there is no teaching or suggestion in Garg regarding how to process two or more failures "that relate to a common fault" as in claim 69, let alone any teaching or suggestion of "not forward[ing] fault messages that relate to a fault for which the fault management unit has already forwarded a fault message."

Garg Does Not Teach or Suggest Receiving "Fault Messages" Generated "by a Switch Portion and by a Service Processor Portion"

Claim 69 recites "a fault management unit configured to receive fault messages" that are "generated" by two possible sources: "by the switch portion and by the service processor portion." In contrast, the "System Controller" is the only element in Garg disclosed to generate "alarm[s]." *See id, e.g.,* at ¶ [0405]. Garg does not disclose that its fault manager receives alarms from any other source. Thus, even assuming *arguendo* that Garg's system controller constitutes one of "the switch portion" or "the service processor portion," Garg would still not teach or suggest "a fault management unit configured to receive fault messages generated by the switch portion **and** by the service processor portion," as recited in claim 69.

Garg Does Not Teach or Suggest "Forward[ing] a Fault Message to [an] External Management Entity"

Claim 69 recites a "fault management unit configured to receive fault messages generated" by "the switch portion" and the "service processor portion" within the "combined switch and service processor module." Claim 69 further recites that the fault management unit "forward[s] a fault message to [an] external management entity" "via [a] data interface" of the "combined switch and service processor module" where the "data interface" is "configured to communicate with [the] external management entity via a network." Claim 69 thus refers to different elements: The "fault management unit" of claim 69 is configured to communicate a "fault message" from a "combined switch and service processor module" "via a network" to an "external management entity." The claim specifies that the "management entity" is external to the "combined switch and service processor."

In contrast, Garg discloses its "Fault Manager module" receiving "alarm[s]" from its "System Controller" and "issu[ing] scForcedSwitchover() command[s]" back to the "System Controller." *See id.* at ¶ [0402]. Thus, even assuming, for example, that the "System Controller" of Garg constitutes one or both portions of the "combined switch and service processor module" of claim 69, Garg does not teach or suggest that its "Fault Manager module" is configured to "forward" a "received message" to a "management entity" that is "external" to the "System Controller" and "Fault Manager Module."

Instead, at best, Garg discloses that its "Fault Manager module" issues "scForcedSwitchover() command[s]" back to the "System Controller." Note that the Examiner cannot contend that Garg's "System Controller" constitutes the "external management entity," as this "management entity" is recited as being "external" to the "combined switch and service processor module" that includes the "switch portion" and the "service processor portion" from which "fault messages" can be received.

***

For at least the reasons stated above, Appellant submits that even assuming a valid motivation to combine the references (which Appellant does not concede), the combination of references proposed by the Examiner would not include each and every feature of claim 69, and therefore cannot be the basis of a *prima facie* case of obviousness with respect to claim 69. Accordingly, the Examiner's rejections of claim 69 and its dependent claims are believed to be in error. The rejections of the other claims in this group are believed to be in error for at least reasons similar to those provided in support of claim 69.

2) Rejection under 35 U.S.C. § 103(a) over Schwartz in view of Garg and Ohkubo et al (U.S. Patent No. 5,276,683).

Claim 81 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Schwartz in view of Garg and Ohkubo et al (U.S. Patent No. 5,276,683). Appellant submits that claim 81 is believed to distinguish over the cited references at least by virtue of its dependency on claim 69. Accordingly, claim 81 stands or falls with representative claim 69.

3) Rejection under 35 U.S.C. 103(a) over Schwartz and Garg in view of Tzeng et al (U.S. Pub. No. 2005/0122825).

Claims 101, 102, and 103 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schwartz and Garg in view of Tzeng et al (U.S. Pub. No. 2005/0122825).

Appellant traverses the rejection of these claims for at least the reasons set forth below. Claims 102 and 103 stand or fall with representative claim 101, which recites as follows:

> 101. The combined switch and service processor module of claim 69, wherein the switch portion is configured to:
>
> detect a fault in an information processing module coupled to the switch portion;
>
> in response to detecting a fault in the information processing module:
>
> disable the network port of the information processing module; and
>
> convey a corresponding fault message to the fault management unit.

The Examiner cites Garg's abstract as teaching "detect[ing] a fault in an information processing module coupled to the switch portion," as recited in claim 101. Final Office Action at 8. The Examiner relies on Tzeng for teaching the remaining portions of the claim. *Id.*

As an initial matter, Appellant disagrees that Garg discloses each of the features alleged by the Examiner. To extent Garg teaches faults, it is within the context of a failing set of resources of an application. *See, e.g.,* Garg ¶ [0046]. Claim 101, however, specifies that the "detect[ed] fault [is] in an information processing module coupled to the switch portion." While Garg discloses that a "resource set refers to a group of resources (such as messages, data, or network elements) that are used by the application," *Id.* at ¶ [0036], Garg does not disclose that it "detect[s] a fault in an information processing module coupled to the switch portion," as recited in claim 101. Appellant thus submits that Garg does not teach or suggest each and every feature alleged by the Examiner.

Furthermore, Tzeng does not disclose each of the features alleged by the Examiner. Tzeng is directed to a "new switching architecture for integrating multiple switche[s] into a single device." Tzeng ¶ [0003]. To this end, Tzeng discloses a network switch that can "control[] the inflow of additional data packets" based on the number of packets that it receives. *See id* at ¶¶ [0136] and [0137]. For example, Tzeng discloses that if a "packet on port 14 can not be transmitted out," Tzeng will issue a "back pressure warning message to port 26" to "stop the traffic from port 26 to port 12." *See id* ¶¶ [0064] and [0351]. Appellant respectfully submits that Tzeng's switch does not teach or suggest several features of claim 101. First, Tzeng's switch does not "disable the network port of [an] information processing module." Rather, it disables a port within itself. *See id.* at Fig. 42 and ¶ [0351]. Second, Tzeng's switch does not "disable [a] network port" "in response to detecting a fault in the information processing module." Rather, the fact that a packet cannot be transmitted by the switch is detected by a unit within the switch—i.e., PMMU on an SOC10. *See id.* at ¶¶ [0350] and [0351]. Appellant therefore submits that Tzeng does not teach or suggest at least these features.

***

For at least the reasons stated above, Appellant submits that even assuming a valid motivation to combine the references (which Appellant does not concede), the combination of references proposed by the Examiner would not include each and every feature of claim 101, and therefore cannot be the basis of a *prima facie* case of obviousness with respect to claim 101. Accordingly, the Examiner's rejections of claim 101 and the other claims in this group are believed to be in error for at least these reasons.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 69-73, 76-80, 82, 86-90, 93-97, and 100-104 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $540.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-85400/DMM.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Appellant hereby petitions for such extension.

Respectfully submitted,

Date: April 5, 2010

By: /Dean M. Munyon/
Dean M. Munyon
Reg. No. 42,914

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P. O. Box 398
Austin, Texas 78767
(512) 853-8800

VIII. CLAIMS APPENDIX

The following lists claims 69-73, 76-82, 86-90, 93-97 and 100-104, incorporating entered amendments, as on appeal.

69. A combined switch and service processor module for a modular computer system, comprising:

a switch portion;

a service processor portion;

a data interface configured to communicate with an external management entity via a network; and

a fault management unit configured to receive fault messages generated by the switch portion and by the service processor portion, including respective fault messages generated by the switch portion and the service processor portion that relate to a common fault, and wherein the fault management unit is further configured to perform processing on a received fault message to determine whether to forward the received message to the external management entity via the data interface, wherein the fault management unit is configured to not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message to the external management entity.

70. The combined switch and service processor module of claim 69, wherein the fault management unit is implemented within the service processor portion.

71. The combined switch and service processor module of claim 69, wherein the fault management unit stores details of fault messages received irrespective of whether the message is forwarded to the external management entity.

72. The combined switch and service processor module of claim 71, wherein the stored details of the fault messages includes data describing an action taken by the originator of the fault message in response to detection of the fault.

73. The combined switch and service processor module of claim 71, wherein the stored details of fault messages are analyzed to determine whether any reversement actions are required by the originator of a given fault message when a fault repair is attempted.

76. The combined switch and service processor module of claim 69, wherein the switch portion and service processor portion are implemented by separate hardware within the module.

77. The combined switch and service processor module of claim 69, wherein the switch portion and service processor portion are implemented by common hardware within the module.

78. The combined switch and service processor module of claim 69, wherein the service processor portion is configured to operate in master/slave relationship with a service processor portion of a further combined switch and service processor module of the modular computer system; and

wherein the service processor portion is further configured to automatically synchronize management information with the service processor portion of the further combined switch and service processor via the data interface in accordance with the master/slave relationship.

79. The combined switch and service processor module of claim 69, wherein the switch and service processor portions are each configured to communicate with the external management entity to obtain a unique address within a computing environment into which the modular computer system is connected.

80. The combined switch and service processor module of claim 69, wherein the service processor portion has a user interface configured to receive and forward communications between the external management entity and the switch portion.

81. The combined switch and service processor module of claim 69, wherein the switch and service processor portions are each configured to create a unique identifier using data unique to the respective portions; and

wherein the service processor portion is configured to supply the service processor portion's unique identifier to the switch.

82. A computer system comprising:

a combined switch and service processor module, comprising:

a switch portion;

a service processor portion;

a data interface configured to communicate with an external management entity via a network; and

a fault management unit configured to receive fault messages generated by the switch portion and by the service processor portion, including respective fault messages generated by the switch portion and the service processor portion that relate to a common fault, and wherein the fault management unit is further configured to perform processing on a received fault message to determine whether to forward the received message to the external management entity via the data interface, wherein the fault management unit is configured to not forward fault messages that relate to a fault for which the fault management unit has already forwarded a fault message to the external management entity.

86. A method of operating a combined switch and service processor module for a modular computer system, the combined switch and service processor module having: a switch portion; a service processor portion; a data interface configured to communicate with an external management entity via a network; and a fault management unit; the method comprising:

the fault management unit receiving fault messages generated by the switch portion and by the service processor portion including a first fault message generated by

the switch portion and a second fault message generated by the service processor portion, wherein the first and second fault messages relate to a common fault;

the fault management unit performing processing on a received fault message to determine whether to forward the received message to the external management entity via the data interface; and

the fault management unit not forwarding fault messages that relate to a fault for which the fault management unit has already forwarded a fault message to the external management entity.

87. The computer system of claim 82, wherein the fault management unit is implemented within the service processor portion.

88. The computer system of claim 82, wherein the fault management unit is further configured to store details of fault messages received irrespective of whether the message is forwarded to the external management entity.

89. The computer system of claim 88, wherein the stored details of the fault messages include data describing an action taken by the originator of the fault message in response to detection of the fault.

90. The computer system of claim 88, wherein the fault management unit is further configured to analyze the stored details of fault messages to determine whether any reversement actions are required by the originator of a given fault message when a fault repair is attempted.

93. (Previously Presented) The computer system of claim 82, wherein the service processor portion is configured to operate in master/slave relationship with a service processor portion of a further combined switch and service processor module of the computer system; and

wherein the service processor portion is further configured to automatically synchronize management information with the service processor portion of the further

combined switch and service processor via the data interface in accordance with the master/slave relationship.

94. The method of claim 86, wherein the fault management unit is implemented within the service processor portion.

95. The method of claim 86, further comprising the fault management unit storing details of fault messages received irrespective of whether the message is forwarded to the external management entity.

96. The method of claim 95, wherein the stored details of the fault messages includes data describing an action taken by the originator of the fault message in response to detection of the fault.

97. The method of claim 95, further comprising the fault management unit analyzing the stored details of fault messages to determine whether any reversement actions are required by the originator of a given fault message when a fault repair is attempted.

100. The method of claim 86, further comprising operating the service processor portion in master/slave relationship with a service processor portion of a further combined switch and service processor module of the modular computer system; and

the service processor portion automatically synchronizing management information with the service processor portion of the further combined switch and service processor via the data interface in accordance with the master/slave relationship.

101. The combined switch and service processor module of claim 69, wherein the switch portion is configured to:

detect a fault in an information processing module coupled to the switch portion;

in response to detecting a fault in the information processing module:

disable the network port of the information processing module; and

convey a corresponding fault message to the fault management unit.

102. The computer system of claim 82, wherein the switch portion is configured to:

detect a fault in an information processing module coupled to the switch portion;

in response to detecting a fault in the information processing module:

disable the network port of the information processing module; and

convey a corresponding fault message to the fault management unit.

103. (Previously Presented) The method of claim 86, further comprising:

the switch portion detecting a fault in an information processing module;

in response to the detecting:

the switch portion disabling the network port of the information processing module; and

the switch portion conveying a corresponding fault message to the fault management unit.

104. The combined switch and service processor module of claim 69, wherein the fault management unit is configured to:

receive a first fault message generated by the switch portion that relates to a particular fault;

forward the first fault message to the external management entity;

receive a second fault message generated by the service portion, wherein the second fault message relates to the particular fault, and wherein the second fault message is received by the fault management unit after receiving the first fault message; and

not forward the second fault message to the external management entity in response to determining that the second fault message is related to the particular fault and that the first fault message has already been forwarded to the external management entity.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.